

Link github : [farrazrayacode/TUGAS-W3-IMPAL-](https://github.com/farrazrayacode/TUGAS-W3-IMPAL-): Tugas Kelompok Impelementasi dan Perencanaan Perangkat Lunak Minggu ke-3

Nama : Yolanda Rahma Chrysti

NIM : 103012300150

Soal 1

PSPEC - Validasi Segitiga :

Problem Specification

Tujuan:

1. Menerima 3 angka sembarang (integer atau pecahan), memutuskan:
2. Apakah tiga sisi membentuk segitiga valid?
3. Jika valid, tentukan semua kategori yang berlaku: Sama Sisi (Equilateral), Sama Kaki (Isosceles), Siku-Siku (Right), atau Bebas (Scalene).
4. Untuk pecahan, memakai presisi/toleransi 1% pada perbandingan sisi dan persegi sisi.
5. Output seluruh diagnosa dengan penjelasan.

Input:

Tiga angka, a, b, c (dapat berupa integer maupun pecahan)

Output:

1. Tidak valid jika:
 - Ada nilai ≤ 0
 - Panjang sisi terbesar \geq jumlah dua lainnya
2. Jika valid, berikan detail:
 - Equilateral (semua sisi sama)
 - Isosceles (ada dua sisi sama)
 - Right (siku-siku, menggunakan toleransi 1% jika bilangan real)
 - Scalene (tidak termasuk tiga kategori di atas)

Code :

```
#include <stdio.h>
#include <math.h>

#define EPSILON 0.01 // Toleransi 1%

int is_almost_equal(double x, double y) {
    if (x == 0 && y == 0) return 1;
    return fabs(x-y) / fmax(fabs(x), fabs(y)) <= EPSILON;
}

void sort_desc(double *x, double *y, double *z) {
    double a = *x, b = *y, c = *z, temp;
    // Sorting agar a ≥ b ≥ c
    if (a < b) { temp = a; a = b; b = temp; }
    if (a < c) { temp = a; a = c; c = temp; }
    if (b < c) { temp = b; b = c; c = temp; }
    *x = a; *y = b; *z = c;
}

void diagnose_triangle(double a, double b, double c, int is_int) {
    sort_desc(&a, &b, &c); // a = terbesar, b = tengah, c = terkecil
```

```

printf("Diagnosa Input: %.10g %.10g %.10g\n", a, b, c);
if ((a <= 0) || (b <= 0) || (c <= 0)) {
    printf("Tidak ada segitiga dapat dibangun (ada sisi ≤ 0).\n");
    return;
}
if (a >= b + c) {
    printf("Tidak ada segitiga dapat dibangun (panjang maksimum tidak < jumlah dua sisi lain).\n");
    return;
}

// Equilateral
int equi = is_int ? (a==b && b==c) : (is_almost_equal(a,b) && is_almost_equal(b,c));
// Isosceles (tapi bukan sama sisi)
int iso = !equi && (is_int ? (a==b || a==c || b==c) : (is_almost_equal(a,b) || is_almost_equal(a,c) ||
is_almost_equal(b,c)));
// Right triangle
int right = is_int ? (a*a == b*b + c*c) : is_almost_equal(a*a, b*b + c*c);

// Output
printf("Jenis Segitiga:\n");
if (equi) printf("- SAMA SISI (Equilateral)\n");
if (iso) printf("- SAMA KAKI (Isosceles)\n");
if (right) printf("- SIKU-SIKU (Right Triangle)\n");
if (!equi && !iso && !right) printf("- BEBAS (Scalene)\n");

// Detail tambahan (ketelitian selisih ≤ 1%%):
if (right && equi) printf("Catatan: Segitiga sama sisi tidak mungkin siku-siku!\n");
if (right && iso) printf("Catatan: Segitiga ini juga merupakan sama kaki siku-siku.\n");
if (equi) printf("Semua sisi saling sama (atau beda ≤ 1%%).\n");
if (iso && !equi) printf("Dua sisi saling sama (atau beda ≤ 1%%).\n");
if (right) printf("Terdapat sudut 90 derajat (atau mendekati).\n");
}

int main() {
    int mode;
    printf("Pilih mode (0=bulat, 1=pecahan): ");
    scanf("%d", &mode);
    if (mode == 0) {
        int a, b, c;
        printf("Input tiga bilangan bulat: ");
        scanf("%d %d %d", &a, &b, &c);
        diagnose_triangle((double)a, (double)b, (double)c, 1);
    } else {
        double a, b, c;
        printf("Input tiga bilangan pecahan: ");
        scanf("%lf %lf %lf", &a, &b, &c);
        diagnose_triangle(a, b, c, 0);
    }
    return 0;
}

```

Nama : Nasywa Kamila
NIM : 103012300105
Soal 1

Kita asumsikan DPPL programnya memiliki proses-proses utama berikut:

1. Menerima 3 input sisi dari pengguna.
2. (PROSES 1) Memvalidasi apakah 3 sisi tersebut bisa membentuk segitiga.
3. (PROSES 2) Mengklasifikasikan jenis segitiga (Sama Sisi, Sama Kaki, atau Bebas).
4. (PROSES 3) Mengecek secara terpisah apakah segitiga tersebut siku-siku.
5. Menampilkan hasil akhir ke pengguna.

PSPEC - Validasi Segitiga

Tugas: Membuat fungsi yang menerima 3 sisi dan mengembalikan true jika valid, dan false jika tidak valid. Logika:

1. Terima 3 angka (sisi a, b, c).
2. Cek apakah ada sisi yang nilainya kurang dari atau sama dengan nol. Jika ya, tidak valid.
3. Cek apakah sisi terpanjang lebih besar atau sama dengan jumlah dua sisi lainnya. Jika ya, tidak valid.
4. Jika lolos semua cek, maka valid.

Code :

```
PROGRAM TentukanSegitiga;

USES Crt;

PROCEDURE TentukanDenganBilanganBulat;
VAR
  a, b, c, temp : LongInt;
  sisi1, sisi2, sisiTerpanjang: LongInt;
  kuadrat1, kuadrat2, kuadratTerpanjang: Int64;

BEGIN
  Writeln('=====');
  Writeln('| Menentukan Jenis Segitiga (Bilangan Bulat) |');
  Writeln('=====');
  Write('Masukkan panjang sisi a: '); Readln(a);
  Write('Masukkan panjang sisi b: '); Readln(b);
  Write('Masukkan panjang sisi c: '); Readln(c);
  Writeln;

  sisi1 := a;
  sisi2 := b;
  sisiTerpanjang := c;

  IF sisi1 > sisi2 THEN BEGIN temp := sisi1; sisi1 := sisi2; sisi2 := temp; END;
  IF sisi2 > sisiTerpanjang THEN BEGIN temp := sisi2; sisi2 := sisiTerpanjang; sisiTerpanjang := temp; END;
  IF sisi1 > sisi2 THEN BEGIN temp := sisi1; sisi1 := sisi2; sisi2 := temp; END;

  IF (sisi1 <= 0) THEN
  BEGIN
    Writeln('HASIL: Tidak ada segitiga dapat dibangun (sisi tidak boleh nol atau negatif).');
  END
  ELSE IF (sisi1 + sisi2 <= sisiTerpanjang) THEN
  BEGIN
    Writeln('HASIL: Tidak ada segitiga dapat dibangun (sisi terpanjang >= jumlah dua sisi lainnya).');
  END
  ELSE
  BEGIN
    IF (sisi1 = sisiTerpanjang) THEN
    BEGIN
      Writeln('HASIL: SEGITIGA SAMA SISI (EQUILATERAL)');
    END
  END
  END
```

```

END
ELSE
BEGIN
    kuadrat1 := sisi1 * sisi1;
    kuadrat2 := sisi2 * sisi2;
    kuadratTerpanjang := sisiTerpanjang * sisiTerpanjang;

    IF (kuadrat1 + kuadrat2 = kuadratTerpanjang) THEN
    BEGIN
        Writeln('HASIL: SEGITIGA SIKU-SIKU (RIGHT TRIANGLE)');
    END
    ELSE IF (sisi1 = sisi2) OR (sisi2 = sisiTerpanjang) THEN
    BEGIN
        Writeln('HASIL: SEGITIGA SAMA KAKI (ISOSCELES)');
    END
    ELSE
    BEGIN
        Writeln('HASIL: SEGITIGA BEBAS');
    END;
END;
END;
END;

PROCEDURE TentukanDenganBilanganPecahan;
CONST
    Epsilon = 0.01;
VAR
    a, b, c, temp : Real;
    sisi1, sisi2, sisiTerpanjang : Real;
    kuadrat1, kuadrat2, kuadratTerpanjang: Real;

BEGIN
    Writeln('=====');
    Writeln('| Menentukan Jenis Segitiga (Bilangan Pecahan) |');
    Writeln('=====');
    Write('Masukkan panjang sisi a: '); Readln(a);
    Write('Masukkan panjang sisi b: '); Readln(b);
    Write('Masukkan panjang sisi c: '); Readln(c);
    Writeln;

    sisi1 := a;
    sisi2 := b;
    sisiTerpanjang := c;

    IF sisi1 > sisi2 THEN BEGIN temp := sisi1; sisi1 := sisi2; sisi2 := temp; END;
    IF sisi2 > sisiTerpanjang THEN BEGIN temp := sisi2; sisi2 := sisiTerpanjang; sisiTerpanjang := temp; END;
    IF sisi1 > sisi2 THEN BEGIN temp := sisi1; sisi1 := sisi2; sisi2 := temp; END;

    IF (sisi1 <= 0) THEN
    BEGIN
        Writeln('HASIL: Tidak ada segitiga dapat dibangun (sisi tidak boleh nol atau negatif).');
    END
    ELSE IF (sisi1 + sisi2 - sisiTerpanjang <= Epsilon) THEN
    BEGIN
        Writeln('HASIL: Tidak ada segitiga dapat dibangun (sisi terpanjang >= jumlah dua sisi lainnya).');
    END
    ELSE
    BEGIN
        IF (Abs(sisi1 - sisiTerpanjang) <= Epsilon) THEN
        BEGIN
            Writeln('HASIL: SEGITIGA SAMA SISI (EQUILATERAL)');
        END
        ELSE
        BEGIN
            kuadrat1 := sisi1 * sisi1;
            kuadrat2 := sisi2 * sisi2;
            kuadratTerpanjang := sisiTerpanjang * sisiTerpanjang;

            IF (Abs((kuadrat1 + kuadrat2) - kuadratTerpanjang) <= Epsilon) THEN
            BEGIN

```

```

    Writeln('HASIL: SEGITIGA SIKU-SIKU (RIGHT TRIANGLE)');
END
ELSE IF (Abs(sisi1 - sisi2) <= Epsilon) OR (Abs(sisi2 - sisiTerpanjang) <= Epsilon) THEN
BEGIN
    Writeln('HASIL: SEGITIGA SAMA KAKI (ISOSCELES)');
END
ELSE
BEGIN
    Writeln('HASIL: SEGITIGA BEBAS');
END;
END;
END;
END;

VAR
pilihan: Char;

BEGIN
REPEAT
    ClrScr;
    Writeln('=====');
    Writeln('| PROGRAM PENENTU JENIS SEGITIGA |');
    Writeln('=====');
    Writeln('| MENU: |');
    Writeln('| 1. Masukkan Bilangan Bulat |');
    Writeln('| 2. Masukkan Bilangan Pecahan (Desimal)|');
    Writeln('| 3. Keluar |');
    Writeln('=====');
    Write('Pilih [1..3]: ');
    pilihan := ReadKey;
    Writeln(pilihan);
    Writeln;

CASE pilihan OF
    '1': BEGIN
        TentukanDenganBilanganBulat;
        Writeln;
        Writeln('Tekan tombol apa saja untuk kembali ke menu...');
        ReadKey;
        END;
    '2': BEGIN
        TentukanDenganBilanganPecahan;
        Writeln;
        Writeln('Tekan tombol apa saja untuk kembali ke menu...');
        ReadKey;
        END;
    '3': Writeln('Terima kasih telah menggunakan program ini. ');
ELSE
    Writeln('Pilihan tidak valid. Silakan coba lagi. ');
    ReadKey;
END;

UNTIL pilihan = '3';
END.

```

Nama : Farraz Raya Abdillah

NIM : 103012300149

Soal 2

PSPEC - Biaya Servis Komputer :

Tujuan :

1. Menerima beberapa input mengenai servis komputer dan menghitung total harga yang harus dibayarkan oleh customer.

Logika :

1. Jika jumlah komputer yang diservis 1 atau 2, maka harga dasarnya \$50 dan tidak ada biaya tambahan
2. Jika jumlah komputer yang diservis dari 3 sampai 10, maka harga dasarnya \$100 dan ada biaya tambahan setiap peripherals (hardware external) yaitu \$10.
3. Jika jumlah komputer yang diservis diatas 10, maka harga dasarnya \$500 dan ada biaya tambahan setiap peripherals yaitu \$10.
4. Jika waktu servis komputer tidak dalam business hour, maka harga dasar menjadi dua kali lipat
5. Jika customer bisa melakukan drop-off dan pick-up sendiri, maka total base fee dipotong setengah.

Input :

1. Integer : jumlahKomputer dan peripherals
2. String : waktuServis
3. Character : dropOff
4. Real(Desimal) : baseFee, tambahanFee, dan totalFee

Code :

```
program BiayaServisKomputer;
uses crt;

var
    jumlahKomputer, peripherals: integer;
    waktuServis: string;
    dropOff: char;
    baseFee, tambahanFee, totalFee: real;

begin
    clrscr;
    write('Berapa Komputer yang perlu diservis? '); readln(jumlahKomputer);
    write('Berapa jumlah peripherals? '); readln(peripherals);
    write('Kapan waktu service? (business/nonbusiness): '); readln(waktuServis);
    write('Bisakah drop-off dan pickup sendiri? (y/n): '); readln(dropOff);

    { Determine base fee and additional fee }
    if (jumlahKomputer = 1) or (jumlahKomputer = 2) then
    begin
        baseFee := 50;
        tambahanFee := 0;
    end
    else if (jumlahKomputer >= 3) and (jumlahKomputer <= 10) then
    begin
        baseFee := 100;
        tambahanFee := peripherals * 10;
    end
    else
    begin
        baseFee := 500;
        tambahanFee := peripherals * 10;
    end
end;
```

```
end;  
  
{ Business hours check }  
if (waktuServis <> 'business') then  
    baseFee := baseFee * 2;  
  
{ Drop-off discount }  
if (dropOff = 'y') or (dropOff = 'Y') then  
    baseFee := baseFee / 2;  
  
totalFee := baseFee + tambahanFee;  
  
writeln('Total harga servis: $', totalFee:0:2);  
end.
```

Nama : Davino Putra Arraya

NIM : 103012300522

Soal 3

PSPEC – Biaya Servis Komputer

Tujuan:

- A. Menerima input berupa nomor akun dari pelanggan.
- B. Memeriksa apakah nomor akun terdapat dalam daftar akun (ACCOUNTS) dan statusnya valid.
- C. Jika valid → menghasilkan invoice berisi nomor akun dan biaya servis.
- D. Jika tidak valid → menghasilkan pesan error.

Input:

- A. Satu data string nomorAkun yang dimasukkan oleh pelanggan.
- B. Daftar ACCOUNTS sudah tersimpan dalam sistem, berisi:
- C. nomorAkun (string)
- D. statusKode (valid/invalid)
- E. biayaServis (angka, bisa pecahan / desimal).

Pre/Post Conditions

A. Precondition 1:

Customer datang dengan nomorAkun yang cocok dengan data di ACCOUNTS dan statusKode = "valid".

→ Postcondition 1: Sistem mengeluarkan invoice.

B. Precondition 2:

nomorAkun tidak ditemukan atau statusKode != "valid".

→ Postcondition 2: Sistem mengeluarkan pesan error.

Code :

```
#include <stdio.h>
#include <string.h>

#define JUMLAH_AKUN 5

typedef struct {
    char nomorAkun[20];
    char statusKode[10];
    double biayaServis;
} Akun;

int main() {
    Akun daftarAkun[JUMLAH_AKUN] = {
        {"A001", "valid", 150000},
        {"A002", "invalid", 0},
        {"A003", "valid", 250000},
        {"A004", "valid", 100000},
        {"A005", "invalid", 0}
    };

    char inputNomor[20];
    int ditemukan = 0;
```



```
printf("Masukkan Nomor Akun: ");
scanf("%s", inputNomor);

for (int i = 0; i < JUMLAH_AKUN; i++) {
    if (strcmp(inputNomor, daftarAkun[i].nomorAkun) == 0) {
        ditemukan = 1;
        if (strcmp(daftarAkun[i].statusKode, "valid") == 0) {
            printf("=== INVOICE ===\n");
            printf("Nomor Akun : %s\n", daftarAkun[i].nomorAkun);
            printf("Biaya Servis : Rp %.2f\n", daftarAkun[i].biayaServis);
        } else {
            printf("ERROR: Status akun tidak valid.\n");
        }
        break;
    }
}

if (!ditemukan) {
    printf("ERROR: Nomor akun tidak ditemukan.\n");
}

return 0;
}
```